The sinc-based DFT solver

Piotr Magierski (Warsaw),
Aurel Bulgac (Seattle), Kenneth Roche (Oak Ridge)

Goal:

Provide an accurate and symmetry free DFT solver.

Status in 2007:

The method

$$\left|\Psi_j^{(n+1)}\right\rangle = e^{-\lambda \hat{H}} \left|\psi_j^{(n)}\right\rangle, \quad j = 1,\ldots,N$$

$$\left|\Psi_j^{(n+1)}\right\rangle \xrightarrow[normalization]{orthogonalization} \left|\psi_j^{(n+1)}\right\rangle$$

λ - imaginary time step

$$\hat{H}\left|\psi_j^{(n)}\right\rangle \xrightarrow{n\to\infty} E_j \left|\psi_j^{(n)}\right\rangle$$

$$e^{-\lambda \hat{H}} = 1 + \sum_{i=1}^{n} \frac{(-\lambda)^i}{i!} \hat{H}^i + O(\lambda^{n+1})$$

Usually used for n=1

Drawbacks: - slow convergence.
- divergent for too large imaginary time step.
- numerically costly for n>1.

Alternative: we look for the expansion of $\exp(-\lambda \hat{H}) = \exp(-\lambda(\hat{T} + \hat{V}))$ in the form of product of exponents.

$$e^{-\lambda \hat{H}} = e^{-\frac{\lambda}{2}\hat{T}} e^{-\lambda \hat{V}} e^{-\frac{\lambda}{2}\hat{T}} + O(\lambda^3),$$ - second order method

$$e^{-\lambda \hat{H}} = e^{-\frac{\lambda}{6}\hat{V}} e^{-\frac{\lambda}{2}\hat{T}} e^{-\frac{2\lambda}{3}\tilde{V}} e^{-\frac{\lambda}{2}\hat{T}} e^{-\frac{\lambda}{6}\hat{V}} + O(\lambda^5),$$ - fourth order method

M. Aichinger
E. Krotscheck,
Comput. Mater. Sci.
34, 188 (2005)

$$\tilde{V} = \hat{V} + \frac{1}{48}\lambda^2 [\hat{V}, [\hat{T}, \hat{V}]]$$ - dominant term is local in space if $\hat{V}$ is local.

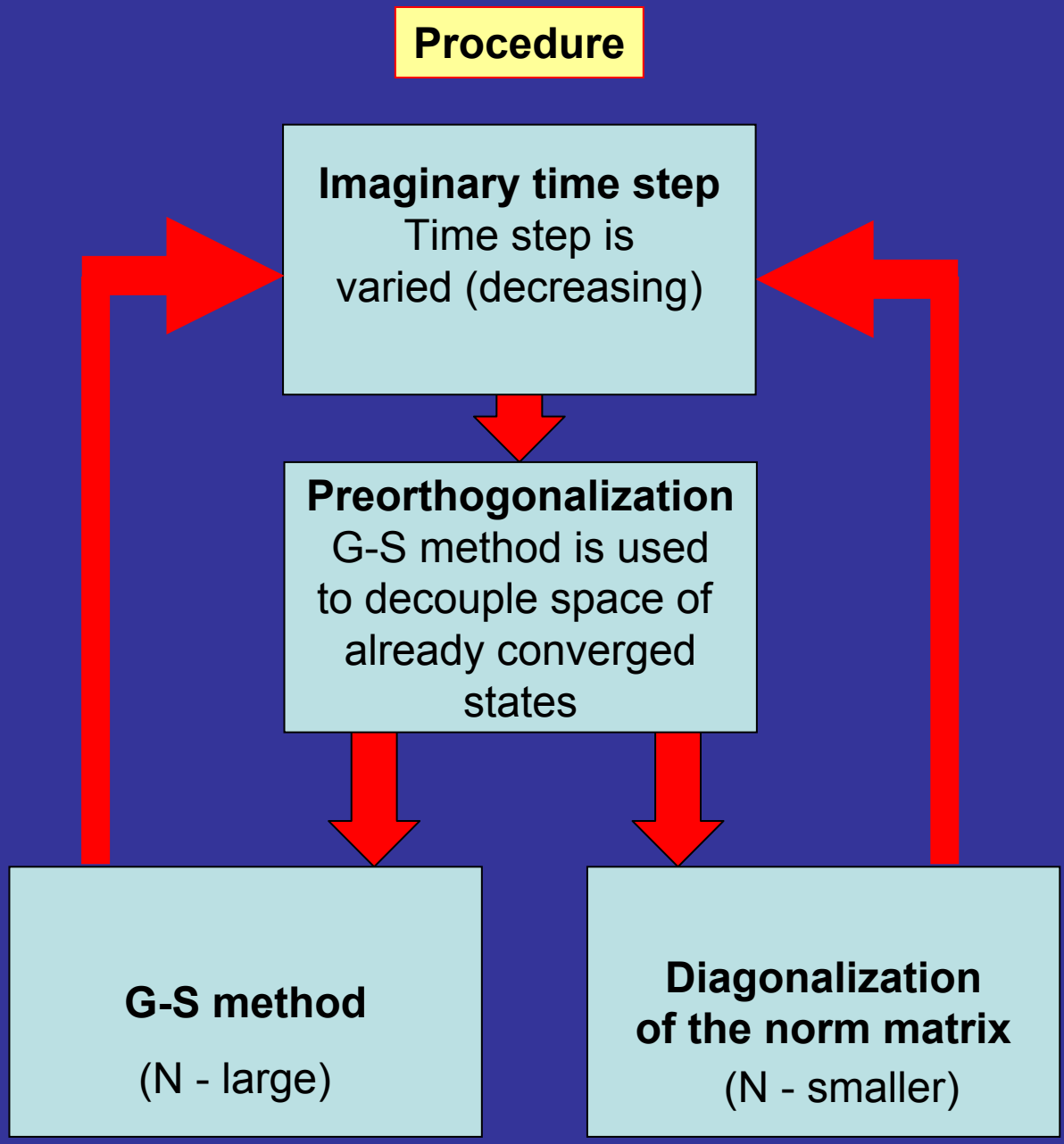
The above expansions are particularly useful if the Hamiltonian can be expressed as a sum of terms either local in the coordinate space (potential), or in the momentum space (kinetic energy).
In such a case one is adviced to use Fast Fourier Transform algorithm during the evolution of the wave function.

$$e^{-\lambda \hat{T}} \psi(\vec{p}) = e^{-\lambda \frac{p^2}{2m}} \psi(\vec{p})$$

$$e^{-\lambda \hat{V}} \psi(\vec{r}) = e^{-\lambda V(\vec{r})} \psi(\vec{r})$$

REAL SPACE ⟺ *FFT* ⟺ **MOMENTUM SPACE**

Advantages:
- Much faster convergence (**order of magnitude** difference between the first order and the second order method).
- The methods **do not diverge** even for large time steps.
- The low cost of **FFT** instead of matrix multiplication.



N – number of evolved wave functions

What has been done during the last year:

- Inclusion of spin-orbit term:

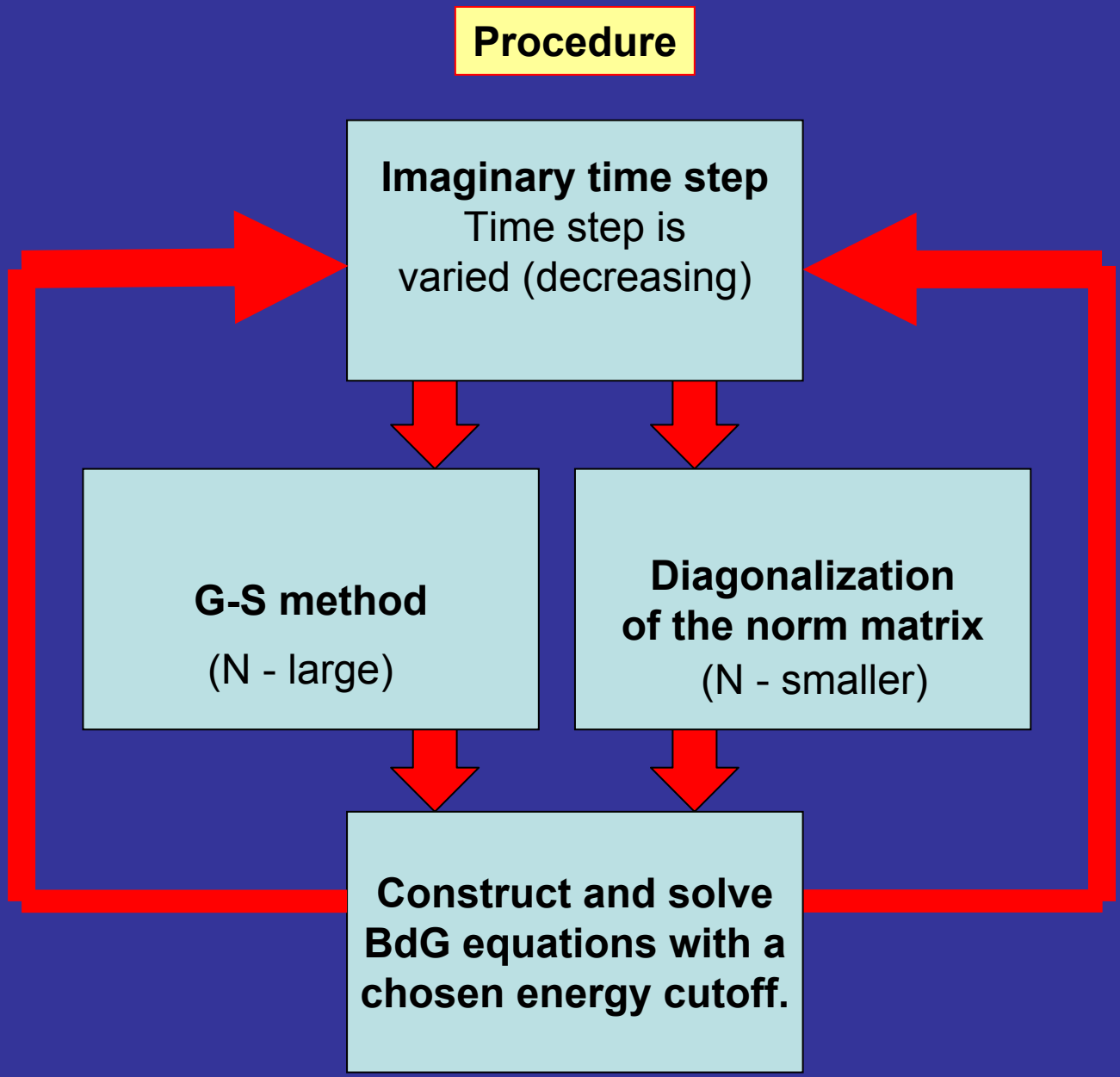
$$e^{-\lambda\hat{H}} \approx e^{-\frac{\lambda}{2}\hat{T}} e^{-\frac{\lambda}{2}\hat{V}} \left(1-\lambda\vec{\nabla}V\cdot(\vec{\sigma}\times\vec{p})+\frac{\lambda^2}{2}\left(\vec{\nabla}V\cdot(\vec{\sigma}\times\vec{p})\right)^2\right) e^{-\frac{\lambda}{2}\hat{V}} e^{-\frac{\lambda}{2}\hat{T}}$$

- Inclusion of pairing:

$$\begin{pmatrix} \hat{h} & \hat{\Delta} \\ \hat{\Delta}^\dagger & -\hat{h}^* \end{pmatrix} \begin{pmatrix} U_n(\vec{r}) \\ V_n(\vec{r}) \end{pmatrix} = E_n \begin{pmatrix} U_n(\vec{r}) \\ V_n(\vec{r}) \end{pmatrix}$$

$$\hat{h} = \begin{pmatrix} -\frac{\hbar^2}{2m_+}\nabla^2 - \mu_+ + V(\vec{r}) & 0 \\ 0 & -\frac{\hbar^2}{2m_-}\nabla^2 - \mu_- - V(\vec{r}) \end{pmatrix} + \lambda\vec{\nabla}V\cdot(\vec{\sigma}\times\vec{p})$$

$$\hat{\Delta} = \begin{pmatrix} 0 & \Delta(\vec{r}) \\ -\Delta(\vec{r}) & 0 \end{pmatrix}; \quad U_n(\vec{r}) = \begin{pmatrix} U_{n+}(\vec{r}) \\ U_{n-}(\vec{r}) \end{pmatrix}; V_n(\vec{r}) = \begin{pmatrix} V_{n+}(\vec{r}) \\ V_{n-}(\vec{r}) \end{pmatrix}$$



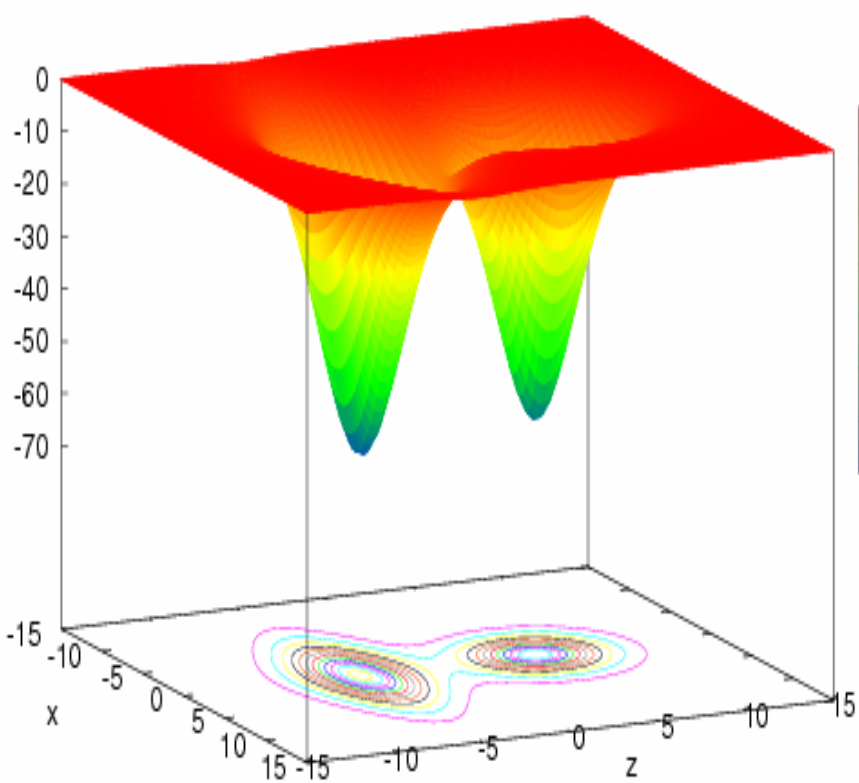
N – number of evolved wave functions

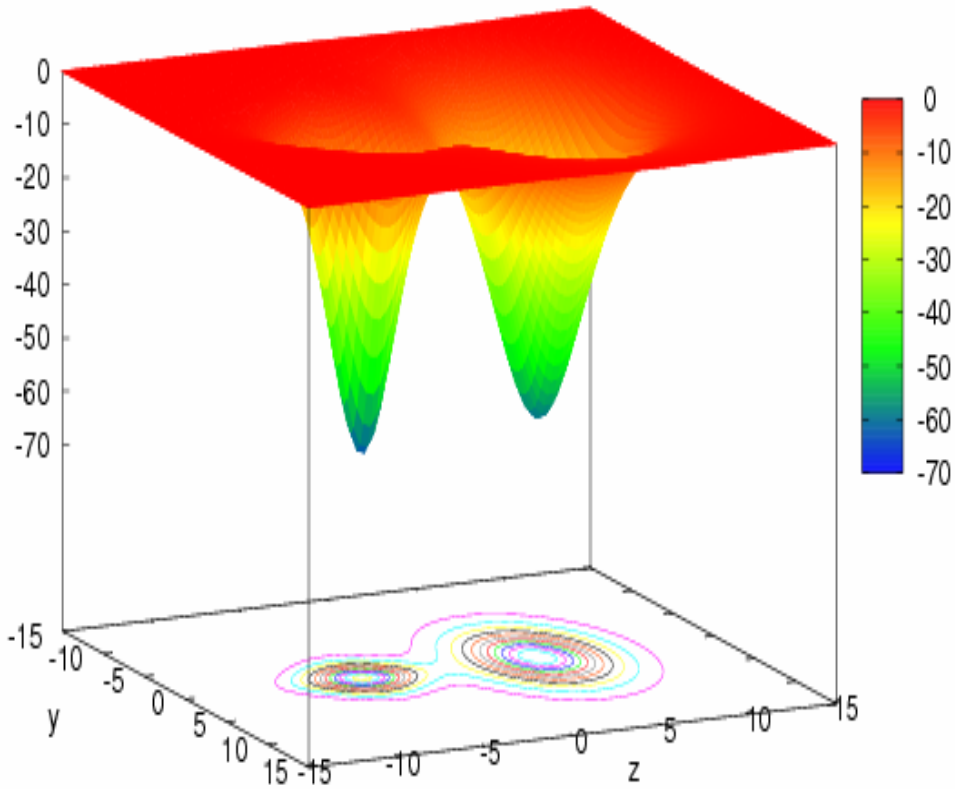
Potential in p-h channel

$$V(\vec{r}) = \frac{V_1}{1+\cosh\left(\frac{r_1}{a}\right)} + \frac{V_2}{\left(1+\cosh\left(\frac{r_{xy}}{b}\right)\right)\left(1+\cosh\left(\frac{z+\xi}{b}\right)\right)}$$

$$r_1 = \sqrt{x^2 + 0.4y^2 + (z-\xi)^2}\,; r_{xy} = \sqrt{0.2x^2 + y^2}$$

$$V_1 = -120MeV\ , V_2 = -250MeV,\ a = 1.5\,fm,\ b = 1.2\,fm,\ \xi = 5\,fm$$



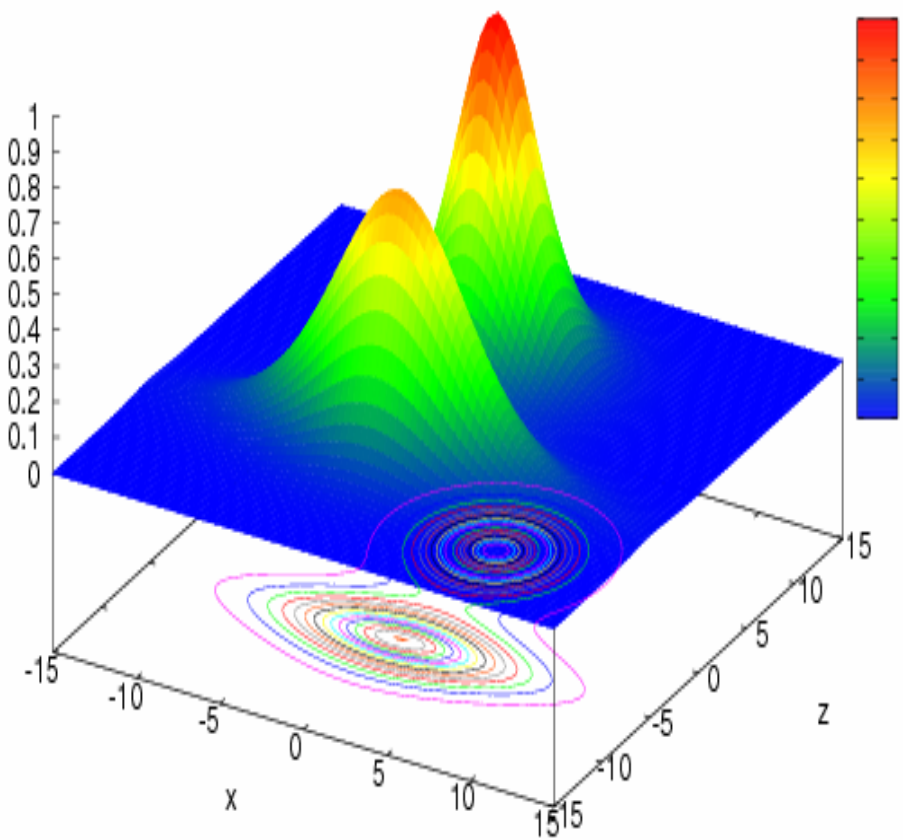


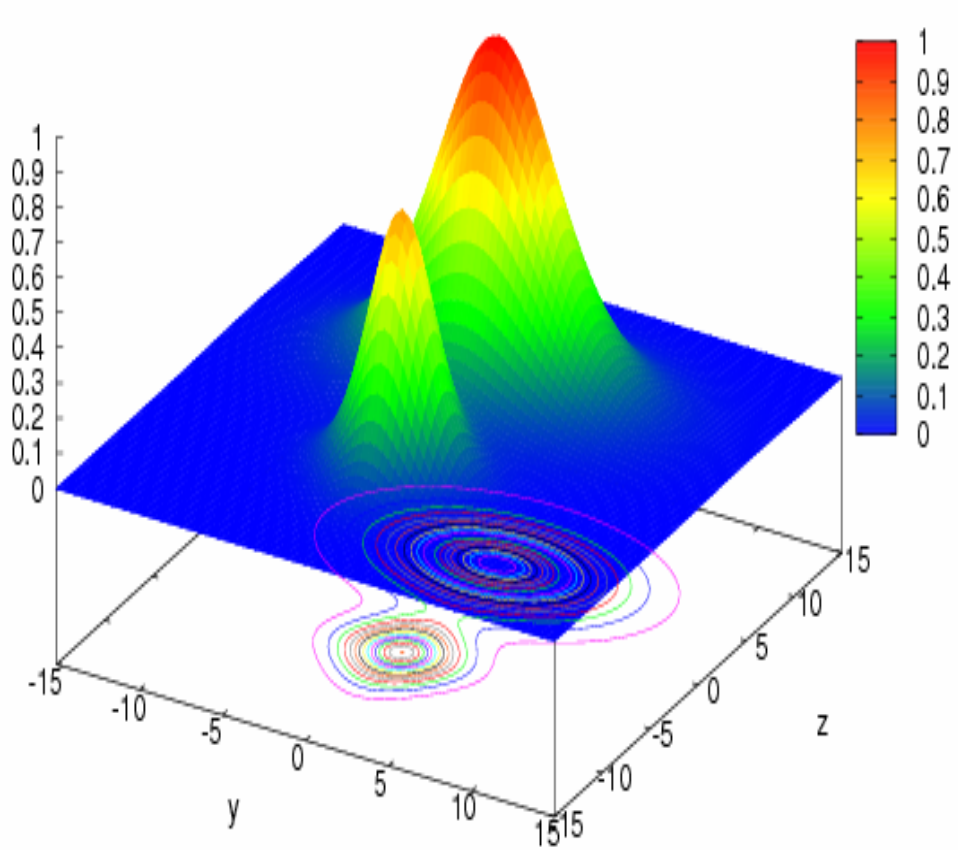
Potential in p-p channel

$$\Delta(\vec{r}) = \frac{\Delta_1}{1+\cosh\left(\frac{r_1}{a}\right)} + \frac{\Delta_2}{\left(1+\cosh\left(\frac{r_{xy}}{b}\right)\right)\left(1+\cosh\left(\frac{z+\xi}{b}\right)\right)}$$

$$r_1 = \sqrt{x^2 + 0.4y^2 + (z-\xi)^2}\,; r_{xy} = \sqrt{0.2x^2 + y^2}$$

$$\Delta_1 = 2MeV\ , \Delta_2 = 3MeV,\ a = 1.5\,fm,\ b = 1.2\,fm,\ \xi = 5\,fm$$



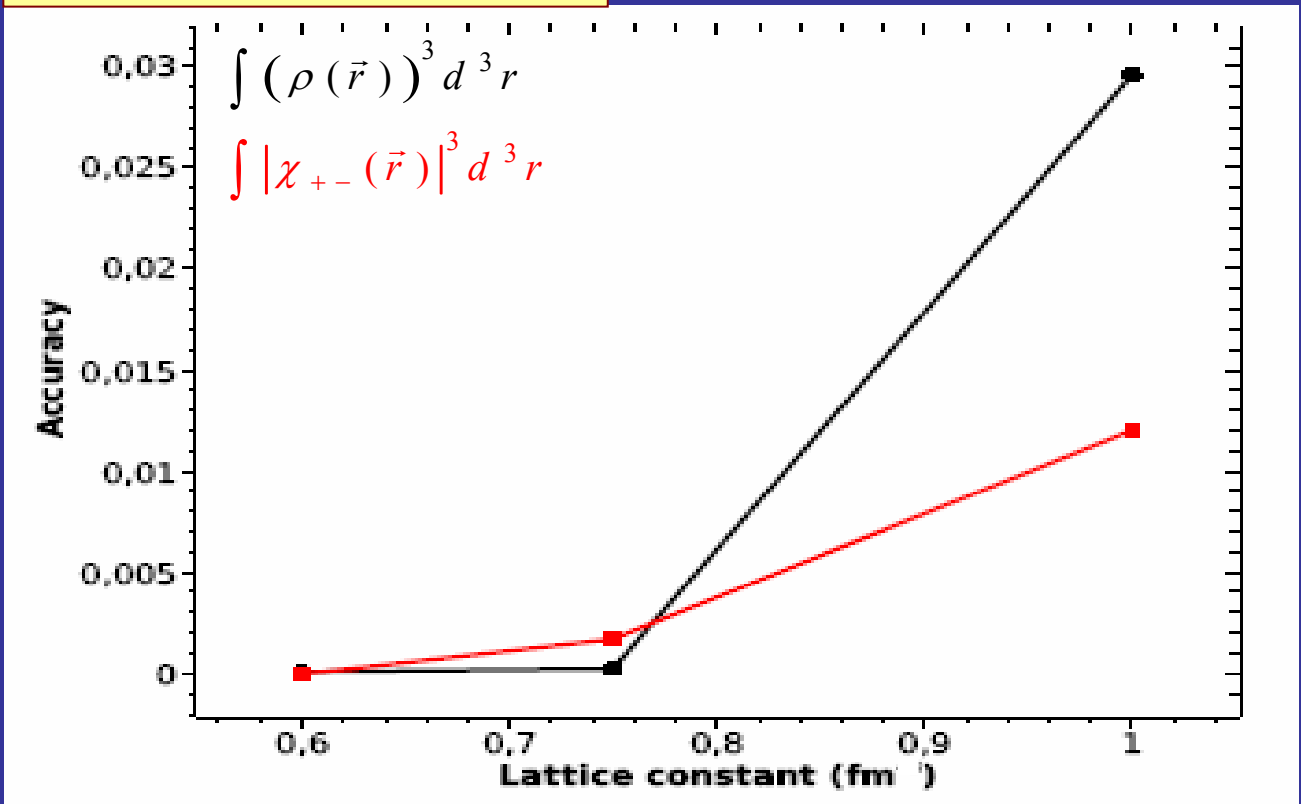


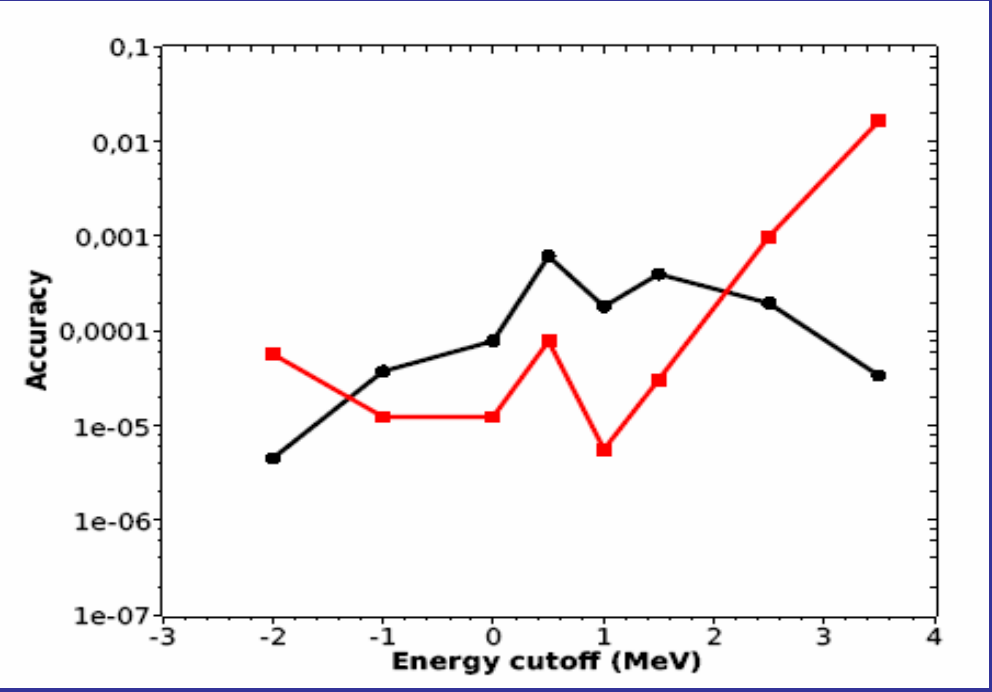
Accuracy (numerical tests)

Number of wave functions: 70
Chemical potential (both spins): -3 MeV (about 40 particles)
Energy cutoff: 1 MeV
Box size: 30 fm

$$\rho_i(\vec{r}) = \sum_n V^*_{ni}(\vec{r}) V^T_{ni}(\vec{r}) \; ; \; i = \pm$$

$$\chi_{+-}(\vec{r}) = \sum_n V^*_{n+}(\vec{r}) U^T_{n-}(\vec{r})$$

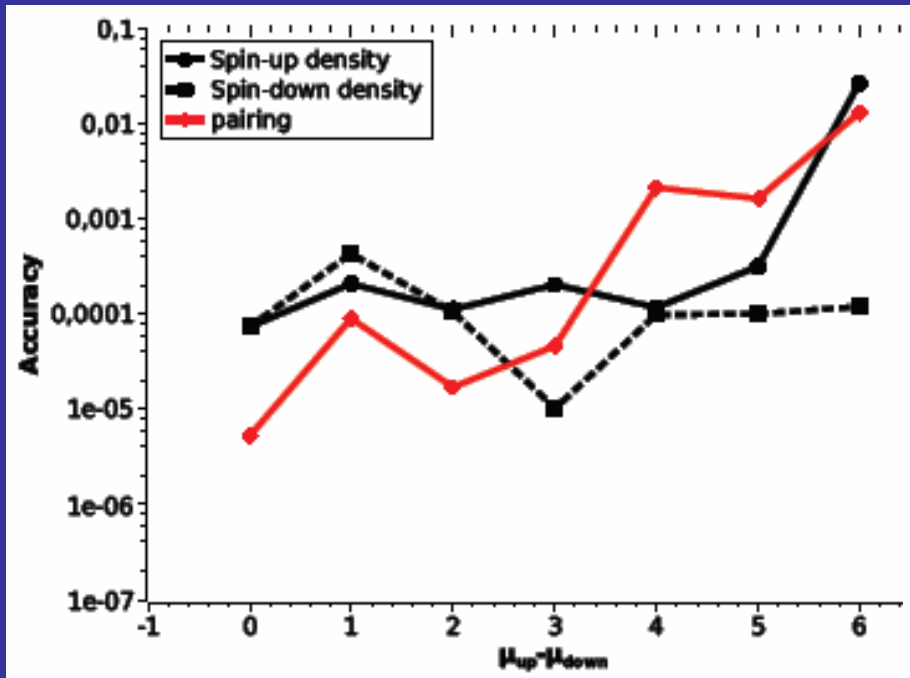


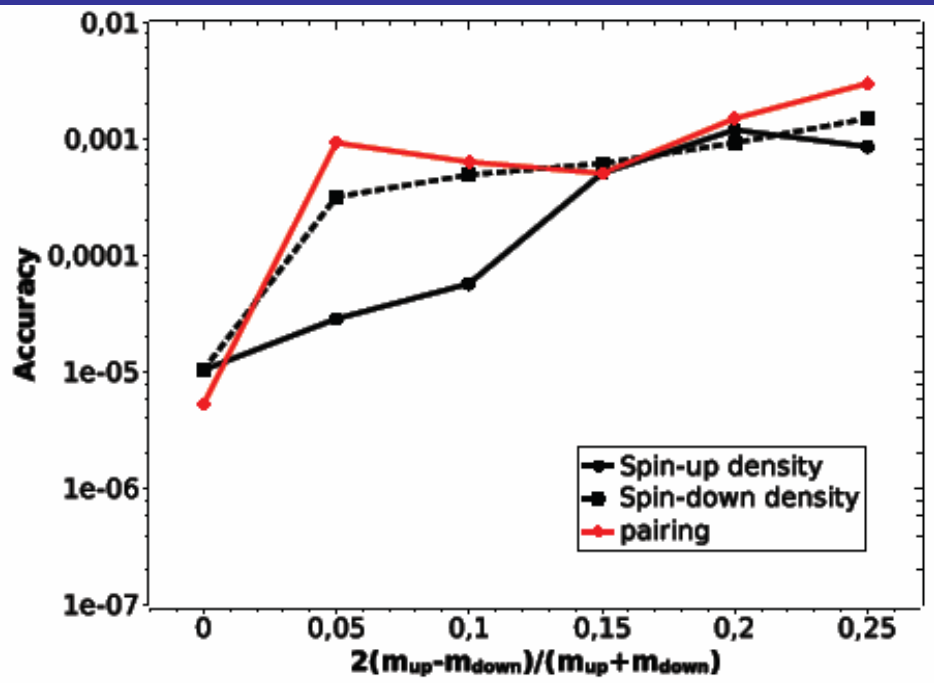


$$\int \left(\rho(\vec{r})\right)^3 d^3 r$$

$$\int \left|\chi_{+-}(\vec{r})\right|^3 d^3 r$$

Box: 30 fm
Lattice constant: 0.6 fm





Summary

At the moment '*the non-selfconsistent DFT solver*' exist.
It can handle arbitrary shape of potentials in p-h and p-p channels.
The accuracy of the results is stable with respect to variations of the potentials, chemical potentials, energy cutoff, effective masses, etc.

1. **Is your Year-2 plan well on track?**
   Absolutely.
2. **What are the aspects of your science that require high-performance computing?**
   Data and computational complexity.
3. **Plan for Year-2 and Year-3:**
   - profile current code.
   - parallelization of the code.
   - self-consistent version.
   - interface between TDSLDA code.